Approximate Consensus in Highly Dynamic Networks: The Role of Averaging Algorithms

Bernadette Charron-Bost, Matthias Függer, Thomas Nowak

Approximate Consensus in Highly Dynamic Networks: The Role of Averaging Algorithms

Bernadette Charron-Bost[1], Matthias Függer[2], and Thomas Nowak[3]

[1] CNRS, École polytechnique charron@lix.polytechnique.fr
[2] Max-Planck-Institut für Informatik mfuegger@mpi-inf.mpg.de
[3] ENS Paris thomas.nowak@ens.fr

Abstract. We investigate the approximate consensus problem in highly dynamic networks in which topology may change continually and unpredictably. We prove that in both synchronous and partially synchronous networks, approximate consensus is solvable if and only if the communication graph in each round has a rooted spanning tree. Interestingly, the class of averaging algorithms, which have the benefit of being memoryless and requiring no process identifiers, entirely captures the solvability issue of approximate consensus in that the problem is solvable if and only if it can be solved using any averaging algorithm.

We develop a proof strategy which for each positive result consists in a reduction to the *nonsplit* networks. It dramatically improves the best known upper bound on the decision times of averaging algorithms and yields a quadratic time non-averaging algorithm for approximate consensus in non-anonymous networks. We also prove that a general upper bound on the decision times of averaging algorithms have to be exponential, shedding light on the price of anonymity.

Finally we apply our results to networked systems with a fixed topology and benign fault models to show that with n processes, up to $2n-3$ of link faults per round can be tolerated for approximate consensus, increasing by a factor 2 the bound of Santoro and Widmayer for exact consensus.

1 Introduction

Recent years have seen considerable interest in the design of distributed algorithms for dynamic networked systems. Motivated by the emerging applications of the Internet and mobile sensor systems, the design of distributed algorithms for networks with a swarm of nodes and time-varying connectivity has been the subject of much recent work. The algorithms implemented in such dynamic networks ought to be decentralized, using local information, and resilient to mobility and link failures.

A large number of distributed applications require to reach some kind of agreement in the network in finite time. For example, processes may attempt to agree on whether to commit or abort the results of a distributed database transaction; or sensors may try to agree on estimates of a certain variable; or

This work has been partially supported by a LIX-DGA contract.

vehicles may attempt to align their direction of motions with their neighbors. Another example is clock synchronization where processes attempt to maintain a common time scale. In the first example, an *exact consensus* is achieved on one of the outcomes (namely, commit or abort) as opposed to the other examples where processes are required to agree on values that are sufficiently close to each other. The latter type of agreement is referred to as *approximate consensus*.

For the exact consensus problem, one immediately faces impossibility results in truly dynamic networks in which some stabilization of the network during a sufficiently long period of time is not assumed (see, e.g., [21] and [19, Chapter 5]). Because of its wide applicability, the approximate consensus problem appears as an interesting weakening of exact consensus to circumvent these impossibility results. The objective of the paper is exactly to study computability and complexity of approximate consensus in dynamic networks in which the topology may change continually and unpredictably.

Dynamic networks. We consider a fixed set of processes that operate in rounds and communicate by broadcast. In the first part of this article, rounds are supposed to be synchronous in the sense that the messages received at some round have been sent at that round. Then we extend our results to partially synchronous rounds with a maximum allowable delay bound.

At each round, the communication graph is chosen arbitrarily among a set of directed graphs that determines the network model. Hence the communication graph can change continually and unpredictably from one round to the next. The local algorithm at each process applies a state-transition function to its current state and the messages received from its incoming neighbors in the current communication graph to obtain a new state.

While local algorithms can be arbitrary in principle, the basic idea is to keep them simple, so that coordination and agreement do not result from the local computational powers but from the flow of information across the network. In particular, we focus on *averaging algorithms* which repeatedly form convex combinations. These algorithms thus have the benefit of requiring little computational overhead, e.g., allowing for efficient implementations, even in hardware. One additional feature of averaging algorithms is to be memoryless in the sense that the next value of each process is entirely determined only from the values of its incoming neighbors in the current communication graph. More importantly, they work in anonymous networks, not requiring processes to have identifiers.

The network model we consider unifies a wide variety of dynamic networks. The most evident class of networks captured by this model are dynamic multi-agent networks, in which communication links frequently go down while other links are established due to the mobility of the agents. The network model can also serve as an abstraction for static or dynamic wireless networks in which collisions and interferences make it difficult to predict which messages will be delivered in time. Finally, it can also be used to model traditional communication networks with a fixed communication graph and some transient link failures.

In our model, the number of processes n is fixed and known to each process. However, all of our results still hold when n is not the exact number of processes but only an upper bound. That allows us to extend the results to a completely dynamic network with a maximal number of processes that may join or leave.

Finally, for simplicity, we assume that all processes start computation at the same round. In fact, it is sufficient to assume that every process eventually participates to the computation either spontaneously (in other words, initiates the computation) or by receiving, possibly indirectly, a message from an initiator.

Contribution. We make the following contributions in this work:

(i) The main result is the exact characterization of the network models in which approximate consensus is solvable. We prove that the approximate consensus problem is solvable in a network model if and only if each communication graph in this model has a rooted spanning tree. This condition guarantees that the network has at least one coordinator in each round. The striking point is that coordinators may continually change over time without preventing nodes from converging to consensus. Accordingly, the network models in which approximate consensus is solvable are called *coordinated network models*. This result highlights the key role played by averaging algorithms in approximate consensus: the problem is solvable if and only if it can be solved using any averaging algorithm.

(ii) With averaging algorithms, we show that agreement with precision of ε can be reached in $O\left(n^{n+1} \log \frac{1}{\varepsilon}\right)$ rounds in a coordinated network model, which dramatically improves the previous bound in [8]. As a matter of fact, every general upper bound for the class of averaging algorithms has to be exponential in the size of the network as exemplified by the *butterfly network* model [11,20]. Besides we derive a non-averaging algorithm that achieves agreement with precision of ε in $O\left(n^2 \log \frac{1}{\varepsilon}\right)$ rounds in non-anonymous networks.

(iii) We extend our computability and complexity results to the case of *partially synchronous rounds* in which communication delays may be non-null but are bounded by some positive integer Δ. We prove the same necessary and sufficient condition on network models for solvability of approximate consensus, and give an $O\left(n^{n\Delta+1} \log \frac{1}{\varepsilon}\right)$ upper bound on the number of rounds needed by averaging algorithms to achieve agreement with precision of ε.

(iv) Finally, as an application of the above results, we revisit approximate consensus in the context of communication faults. We prove a new result on the solvability of approximate consensus in a complete network model in the presence of benign communication faults, which shows that the number of link faults that can be tolerated increases by a factor 2 when solving approximate consensus instead of consensus.

Related work. Agreement problems have been extensively studied in the framework of static communication graphs or with limited topology changes (see, e.g., [19,3]). In particular, the approximate consensus problem has been studied in numerous papers in the context of a complete graph and at most f faulty processes (see, e.g., [14,15,2]). In the case of benign failures, this yields communication graphs with a *fixed* core of at least $n - f$ processes that have outgoing links to all processes, and so play the role of steady coordinators of the network.

There is also a large body of previous work on general dynamic networks. However, in much of them, topology changes are restricted and the sequences of communication graphs are supposed to be "well-formed" in various senses. Such well-formedness properties are actually opposite to the idea of unpredictable changes. In [1], Angluin, Fischer, and Jiang study the *stabilizing consensus problem* in which nodes are required to agree exactly on some initial value, but without necessarily knowing when agreement is reached, and they assume that any two nodes can directly communicate infinitely often. In other words, they suppose the limit graph formed by the links that occur infinitely often to be complete. To solve the consensus problem, Biely, Robinson, and Schmid [5] assume that throughout every block of $4n - 4$ consecutive communication graphs there exists a stable set of roots. Coulouma and Goddard [12] weaken the latter stability condition to obtain a characterization of the sequences of communication graphs for which consensus is solvable. Kuhn, Lynch, and Oshman [17] study variations of the *counting* problem; they assume bidirectional links and a stability property, namely the T-*interval connectivity* which stipulates that there exists a stable spanning tree over every T consecutive communication graphs. All their computability results actually hold in the case of 1-interval connectivity which reduces to a property on the *set* of possible communication graphs, and the cases $T > 1$ are investigated just to improve complexity results. Thus they fully model unpredictable topology changes, at least for computability results on counting in a dynamic network.

The network model in [17], however, assumes a static set of nodes and communication graphs that are all bidirectional and connected. The same assumptions are made to study the time complexity of several variants of consensus [18] in dynamic networks. Concerning the computability issue, such strong assumptions make exact agreement trivially solvable: since communication graphs are continually strongly connected, nodes can collect the set of initial values and then make a decision on the value of some predefined function of this set.

The most closely related pieces of work are without a doubt those about asymptotic consensus and more specifically *consensus sets*: a consensus set is a set of stochastic matrices such that every infinite backward product of matrices from this set converges to a rank one matrix. Computations of averaging algorithms correspond to infinite products of stochastic matrices, and the property of asymptotic consensus is captured by convergence to a rank one matrix. Hence, when an upper bound on the number of nodes is known, the notion of network models in which approximate consensus is solvable reduces to the notion of consensus sets if we restrict ourselves to averaging algorithms. However the characterization of consensus sets in [13,6] is not included into our main computability result for approximate consensus since the fundamental assumption of a self loop at each node in communication graphs (a process can obviously communicate with itself) does not necessarily hold for the directed graphs associated to stochastic matrices in a consensus set. The characterization of compact consensus sets in [13,6] and our computability result of approximate consensus are thus incomparable.

In the same vein, some of our positive results can be shown equivalent to results about stochastic matrix products in the vast existing literature on asymptotic consensus. Notably Theorem 3 is similar to the central result in [7], but for that we develop a new proof strategy which consists in a reduction to *nonsplit network models*. The resulting proof is much simpler and direct as it requires neither star graphs [7] nor Sarymsakov graphs [23]. Moreover our proof yields a significantly better upper bound on the time complexity of averaging algorithms in coordinated network models, namely $O\big(n^{n+1}\log\frac{1}{\varepsilon}\big)$ instead of $O\big(n^{n^2}\log\frac{1}{\varepsilon}\big)$ in [8]. It also yields a non-averaging algorithm that achieves agreement with precision of ε in $O\left(n^2\log\frac{1}{\varepsilon}\right)$ rounds in non-anonymous networks.

2 Approximate consensus and averaging algorithms

We assume a distributed, round-based computational model in the spirit of the Heard-Of model [10]. A system consists of a set of processes $[n] = \{1,\ldots,n\}$. Computation proceeds in *rounds*: In a round, each process sends its state to its outgoing neighbors, receives values from its incoming neighbors, and finally updates its state. The value of the updated state is determined by a deterministic algorithm, i.e., a transition function that maps the values in the incoming messages to a new state value. Rounds are communication closed in the sense that no process receives values in round k that are sent in a round different from k.

Communications that occur in a round are modeled by a directed graph $G = ([n], E(G))$ with a self-loop at each node. The latter requirement is quite natural as a process can obviously communicate with itself instantaneously. Such a directed graph is called a *communication graph*. We denote by $\mathrm{In}_p(G)$ the set of incoming neighbors of p and by $\mathrm{Out}_p(G)$ the set of outgoing neighbors of p in G. Similarly $\mathrm{In}_S(G)$ and $\mathrm{Out}_S(G)$ denote the sets of the incoming and outgoing neighbors of the nodes in a non-empty set $S \subseteq [n]$. The cardinality of $\mathrm{In}_p(G)$ is called the *in-degree* of process p in G.

A *communication pattern* is a sequence $(G(k))_{k\geqslant 1}$ of communication graphs. Here, $E(k)$, $\mathrm{In}_p(k)$ and $\mathrm{Out}_p(k)$ stand for $E\left(G(k)\right)$, $\mathrm{In}_p(G(k))$ and $\mathrm{Out}_p(G(k))$, respectively.

Each process p has a *local state* s_p whose value at the end of round $k \geqslant 1$ is denoted by $s_p(k)$. Process p's initial state, i.e., its state at the beginning of round 1, is denoted by $s_p(0)$. The *global state* at the end of round k is the collection $s(k) = (s_p(k))_{p\in[n]}$. The *execution* of an algorithm from global initial state $s(0)$, with communication pattern $(G(k))_{k\geqslant 1}$ is the unique sequence $(s(k))_{k\geqslant 0}$ of global states defined as follows: for each round $k \geqslant 1$, process p sends $s_p(k-1)$ to all the processes in $\mathrm{Out}_p(k)$, receives $s_q(k-1)$ from each process q in $\mathrm{In}_p(k)$, and computes $s_p(k)$ from the incoming messages, according to the algorithm's transition function.

Consensus and approximate consensus. A crucial problem in distributed systems is to achieve agreement among local process states from arbitrary initial local states. It is a well-known fact that this goal is not easily achievable in the context of dynamic network changes [16,21], and restrictions on communication patterns are required for that. A *network model* thus is a non-empty set $\mathcal{N}$ of communication graphs, those that may occur in communication patterns.

We now consider the above round-based algorithms in which the local state of process p contains two variables x_p and dec_p. Initially, the range of x_p is $[0,1]$ and $dec_p = \perp$ (which informally means that p has not decided). Process p is allowed to set dec_p to the current value of x_p, and so to a value v different from $\perp$, only once; in that case we say that p *decides* v. An algorithm *achieves consensus with communication pattern* $(G(k))_{k\geqslant 1}$ if each execution from a global initial state as specified above and with the communication pattern $(G(k))_{k\geqslant 1}$ fulfills the following three conditions: *(i) Agreement:* The decision values of any two processes are equal. *(ii) Integrity:* The decision value of any process is an initial value. *(iii) Termination:* All processes eventually decide.

An algorithm *solves consensus* in a network model $\mathcal{N}$ if it achieves consensus with each communication pattern formed with graphs all in $\mathcal{N}$. Consensus is *solvable in* $\mathcal{N}$ if there exists an algorithm that solves consensus in $\mathcal{N}$. Observe that consensus is solvable in $n-1$ rounds if each communication graph is strongly connected. The following impossibility result due to Santoro and Widmayer [21], however, shows that network models in which consensus is solvable are highly constrained: consensus is not solvable in some "almost complete" graphs. Namely that consensus is not solvable in the network model comprising all communication graphs in which at least $n-1$ processes have outgoing links to all other processes. The above theorem has been originally stated in the context of link faults in a complete communication graph but its scope can be trivially extended to dynamic communication networks.

To circumvent the impossibility of consensus even in such highly restricted network models, one may weaken Agreement into *ε-Agreement:* The decision values of any two processes are within an *a priori* specified $\varepsilon > 0$; and replace Integrity by *Validity:* All decided values are in the range of the initial values of processes.

An algorithm *achieves ε-consensus with communication pattern* $(G(k))_{k\geqslant 1}$ if each execution from a global initial state as specified above and with the communication pattern $(G(k))_{k\geqslant 1}$ fulfills Termination, Validity, and ε-Agreement. An algorithm *solves approximate consensus* in a network model $\mathcal{N}$ if for any $\varepsilon > 0$, it achieves ε-consensus with each communication pattern formed with graphs all in $\mathcal{N}$. Approximate consensus is *solvable in a network model* $\mathcal{N}$ if there exists an algorithm that solves approximate consensus in $\mathcal{N}$.

Averaging algorithms. We focus on *averaging algorithms* which require little computational overhead and, more importantly, have the benefit of working in anonymous networks. The update rules for each variable x_p are of the form:

$$x_p(k) = \sum_{q\in \mathrm{In}_p(k)} w_{qp}(k)\, x_q(k-1), \tag{1}$$

where $w_{qp}(k)$ are positive reals and $\sum_{q\in \mathrm{In}_p(k)} w_{qp}(k) = 1$. In other words, at each round k, process p updates x_p to some weighted average of the values $x_q(k-1)$ it has just received. For convenience, we let $w_{qp}(k) = 0$ if $q \notin In_p(k)$.

An *averaging algorithm with parameter* $\varrho > 0$ is an averaging algorithm with the positive weights uniformly lower bounded by ϱ: $\forall k \geqslant 1,\, p,q \in [n]$:

$w_{qp}(k) \in \{0\} \cup [\varrho, 1]$. Since we strive for distributed implementations of averaging algorithms, $w_{qp}(k)$ is required to be locally computable. Finally note that the decision rule is not specified in the above definition: the decision time immediately follows from the number of rounds that is proven to be sufficient to reach ε-Agreement.

Some averaging algorithms with locally computable weights are of particular interest, such as the *equal neighbor averaging algorithm*, where at each round k process p chooses $w_{qp}(k) = 1/|\operatorname{In}_p(k)|$ for every q in $\operatorname{In}_p(k)$. It is clearly an averaging algorithm with parameter $\varrho = 1/n$.

3 Solvability and complexity of approximate consensus

In this section, we characterize the network models in which approximate consensus is solvable. First we prove that every averaging algorithm solves approximate consensus in *nonsplit network models*, and extend this result to *coordinated network models* by a reduction to the nonsplit case. The latter result which is quite intuitive in the case of a fixed coordinator, actually holds when coordinators vary over time. Our proof of this known result (in the context of products of stochastic matrices) yields a new upper bound on the decision times of averaging algorithms and a quadratic time approximate consensus algorithm for non-anonymous coordinated networks. A classical partitioning argument combined with a characterization of rooted graphs [9] shows that the condition of rooted graphs is actually necessary to solve approximate consensus.

Nonsplit network model. A directed graph G is *nonsplit* if for all processes (p, q), it holds that $\operatorname{In}_p(G) \cap \operatorname{In}_q(G) \neq \emptyset$. A *nonsplit network model* is a network model in which each communication graph is nonsplit.

Intuitively, the occurrence of a nonsplit communication graph makes the variables x_p in an averaging algorithm to come closer together: any two processes p and q have at least one common incoming neighbor, leading to a common term in both p's and q's average. The convergence proof in [9] of infinite backward products of scrambling stochastic matrices, using the sub-multiplicativity of the Dobrushin's coefficient, formalizes this intuition and yields:

Theorem 1 *In a nonsplit network model of n processes, every averaging algorithm with parameter ϱ achieves ε-consensus in $\frac{1}{\varrho} \log \frac{1}{\varepsilon}$ rounds.*

Theorem 1 can be easily extended with respect to the granularity at which the assumption of nonsplit graphs holds. Let the *product* of two directed graphs G and H with the same set of nodes V be the directed graph $G \circ H$ with set of nodes V and a link from (p, q) if there exists $r \in V$ such that $(p, r) \in E(G)$ and $(r, q) \in E(H)$. For any positive integer K, we say a network model $\mathcal{N}$ is K-*nonsplit* if any product of K graphs from $\mathcal{N}$ is nonsplit.

Corollary 2 *In a K-nonsplit network model of n processes, every averaging algorithm with parameter ϱ achieves ε-consensus in $K\varrho^{-K} \log \frac{1}{\varepsilon} + K - 1$ rounds.*

Coordinated network model. A directed graph G is said to be p-*rooted*, for some node p, if for every node q, there exists a directed path from p to q. Then p is called a *root of* G.

While communication graphs remain p-rooted, process p can play the role of network coordinator: its particular position in the network allows p to impose its value on the network. Accordingly, a network model is said to be *coordinated* if each of its graphs is rooted. It is easy to grasp why in the case of a steady coordinator, processes converge to a common value and so achieve approximate consensus when running an averaging algorithm. We now show that the same still holds when coordinators change over time.

Theorem 3 *In a coordinated network model of* n *processes, every averaging algorithm with parameter* ϱ *achieves* ε*-consensus in* $n\varrho^{-n}\log\frac{1}{\varepsilon} + n - 1$ *rounds.*

The following lemma is the heart of our proof. Corollary 2 allows us to conclude.

Lemma 4 *Every coordinated network model with* n *processes is* $(n-1)$*-nonsplit.*

Proof. Let $H_1, \ldots, H_{n-1}$ be a sequence of $n-1$ communication graphs, each of which is rooted. We recursively define the sets $S_p(k)$ by

$$S_p(0) = \{p\} \quad \text{and} \quad S_p(k) = \mathrm{In}_{S_p(k-1)}(H_k) \text{ for } k \in \{1, \ldots, n-1\}\,. \tag{2}$$

Then $S_p(k) = \mathrm{In}_p(H_k \circ \cdots \circ H_1)$; because of the self-loops, $S_p(k) \subseteq S_p(k+1)$ and none of the sets $S_p(k)$ is empty.

Now we have to show that for any $p, q \in [n]$,

$$S_p(n-1) \cap S_q(n-1) \neq \emptyset\,. \tag{3}$$

If $p = q$, then (3) trivially holds. Otherwise, assume by contradiction that (3) does not hold; for each $k \in \{0, \ldots, n-1\}$, the sets $S_p(k)$ and $S_q(k)$ are disjoint. Consider the sequences $S_p(0) \subseteq \cdots \subseteq S_p(n-1)$, $S_q(0) \subseteq \cdots \subseteq S_q(n-1)$, and $S_p(0) \cup S_q(0) \subseteq \cdots \subseteq S_p(n-1) \cup S_q(n-1)$. Because $|S_p(0) \cup S_q(0)| \geqslant 2$ if $p \neq q$ and $|S_p(n-1) \cup S_q(n-1)| \leqslant n$, the latter sequence cannot be strictly increasing by the pigeonhole principle. Therefore $S_p(\ell) \cup S_q(\ell) = S_p(\ell+1) \cup S_q(\ell+1)$ for some $\ell \in \{0, \ldots, n-2\}$. Since $S_p(\ell) \cap S_q(\ell) = \emptyset$ and $S_p(\ell+1) \cap S_q(\ell+1) = \emptyset$, it follows that $S_p(\ell) = S_p(\ell+1)$ and $S_q(\ell) = S_q(\ell+1)$. Hence both $S_p(\ell)$ and $S_q(\ell)$ have no incoming links in the graph $H_{\ell+1}$. This implies these sets both contain all the roots of $H_{\ell+1}$, a contradiction to the disjointness assumption.

The major difference with the previous proofs of this result [7,9] lies in the fact that we deal with "cumulative graphs" which are just nonsplit (scrambling matrices) instead of being star graphs (matrices with a positive column). In other words, we analyze the evolution of the lines and not of the columns of backward products of stochastic matrices. That allows for a drastic improvement of the decision time of averaging algorithms.

From [9], we derive that a directed graph G is rooted iff the acyclic condensation of G has a sole source. Combined with a simple partitioning argument, we show there exists no algorithm, whether or not it is an averaging algorithm, achieving approximate consensus in a network model with some non-rooted graphs. With our positive result in Theorem 3, this gives:

Theorem 5 *The approximate consensus problem is solvable in a synchronous network model $\mathcal{N}$ if and only if $\mathcal{N}$ is a coordinated model.*

Time complexity of approximate consensus. Even with the improvement of Theorem 3, the upper bound on the decision times of averaging algorithms is exponential in the number n of processes. In particular, the equal neighbor averaging algorithm achieves ε-consensus in $O\left(n^{n+1} \log \frac{1}{\varepsilon}\right)$ rounds. The *butterfly network* model [11,20] is an example of a coordinated network model for which the equal neighbor averaging algorithm exhibits an exponentially large decision time. The example does not even require the network to be dynamic, using a time-constant network only. More precisely by spectral gap arguments, we show the following lower bound.

Theorem 6 *There is a coordinated model consisting of one graph such that, for any $\varepsilon > 0$, the equal neighbor averaging algorithm does not achieve ε-consensus by round k if $k = O\big(2^{n/3} \log \frac{1}{\varepsilon}\big)$.*

Another benefit of Lemma 4 is to provide an approximate consensus algorithm with a quadratic decision time. Indeed Lemma 4 corresponds to a *uniform translation* in the Heard-Of model [10] that transforms each block of $n-1$ consecutive coordinated rounds into one nonsplit macro-round. If each process applies an equal neighbor averaging procedure only at the end of each macro-round instead of applying it round by round, the resulting distributed algorithm (cf. Algorithm 1), which is no more an averaging algorithm and requires a unique identifier for each process, achieves ε-consensus in only $O\left(n^2 \log \frac{1}{\varepsilon}\right)$ rounds, hinting at a price of of anonymity.

Algorithm 1 A quadratic time Approximate Consensus algorithm

Initially:
$x_p \in [0,1]$ and $V_p \leftarrow \{(p, x_p)\}$
In round $k \geqslant 1$ do:
send V_p to all processes in $\mathrm{Out}_p(k)$ and receive V_q from all processes q in $\mathrm{In}_p(k)$
$V_p \leftarrow \bigcup_{q \in \mathrm{In}_p(k)} V_q$
if $k \equiv 0 \mod n-1$ **then**
    $x_p \leftarrow \sum_{(q,x_q) \in V_p} w_{qp}(k) x_q$
    $V_p \leftarrow \{(p, x_p)\}$
end if

4 Synchronism and Faults

Partially synchronous networks. Rounds so far have been supposed to be synchronous: messages are delivered in the same round in which they are sent.

In [22,4], the latter condition is relaxed by allowing processes to receive *outdated* messages, and by bounding the number of rounds between the sending and the receipt of messages by some positive integer Δ. That results in the definitions of Δ-*partially synchronous rounds* and Δ-*bounded executions*. The communication graph at round k is understood to be the directed graph defined by the incoming messages at round k. In the case of averaging algorithms, $x_p(k) = \sum_{q \in \mathrm{In}_p(k)} w_{qp}(k)\, x_q\left(\kappa_q^p(k)\right)$, where $k - \Delta \leqslant \kappa_q^p(k) \leqslant k - 1$. Since process p has immediate access to x_p, we assume $\kappa_p^p(k) = k - 1$.

We now extend the results in the previous section to partially synchronous rounds. Our proof strategy is based on a reduction to the synchronous case: each process corresponds to a set of Δ virtual processes, and every Δ-bounded execution of an averaging algorithm with n processes coincides with a synchronous execution of an averaging algorithm with $n\Delta$ processes.

Unfortunately, Theorem 3 does not simply apply since the key property of a self-loop at each node is not preserved in this reduction. We overcome this difficulty by using Corollary 2 directly: First we prove that if all the graphs in the Δ-bounded execution are rooted, then each cumulative graph over $n\Delta$ consecutive rounds of the synchronous execution is nonsplit. To conclude, we observe that Corollary 2 holds even when some nodes have no self-loop. Again the reduction to nonsplit rounds allows for a much better upper bound on the decision time of the equal neighbor algorithm, namely $O\left(n^{n\Delta+1} \log \frac{1}{\varepsilon}\right)$ instead of $O\left(n^{(\Delta n)^2} \log \frac{1}{\varepsilon}\right)$ in [8,9].

We can now extend the characterization of the network models in which approximate consensus is solvable in Theorem 5 to computations with partially synchronous rounds.

Theorem 7 *The approximate consensus problem is solvable in a partially synchronous network model $\mathcal{N}$ if and only if $\mathcal{N}$ is a coordinated model.*

Communication faults. Time varying communication graphs may result from benign communication faults (message losses) in a fixed network. In the light of Theorem 3, we revisit the problem of approximate consensus in the context of a complete network and communication faults.

Theorem 8 *Approximate consensus is solvable in a complete network with n processes if there are at most $2n - 3$ link faults per round.*

Proof. We actually prove that any directed graph with n nodes and at least $n^2 - 3n + 3$ links is rooted. Since $n^2 - 3n + 3 = (n^2 - n) - (2n - 3)$, the theorem immediately follows.

Assume that G is not a rooted graph. Then the condensation of G has two nodes without incoming link. We denote the corresponding two strongly connected components in G by S_1 and S_2, and their cardinalities by n_1 and n_2, respectively. Therefore the number of links in G that are not self-loops is at most equal to $n^2 - n - n_1(n - n_1) - n_2(n - n_2)$. Since $n^2 - n - n_1(n - n_1) - n_2(n - n_2) \leqslant n^2 - 3n + 2$ when $n_1, n_2 \in [n - 1]$, G has at most $n^2 - 3n + 2$ links.

Compared with the impossibility result established by Santoro and Widmayer [21] for exact consensus with $n-1$ faults per round, the above theorem shows that the number of link faults that can be tolerated increases by a factor 2 when solving approximate consensus instead of consensus. Besides it is easy to construct a non-rooted communication graph with n^2-2n+2 links which, combined with Theorem 5, shows that the bound in the above theorem is tight.

5 Discussion

The main goal of this paper has been to characterize the dynamic network models in which approximate consensus is solvable. As for exact consensus, approximate consensus does not require strong connectivity and it can be solved under the sole assumption of rooted communication graphs. However contrary to the condition of a stable set of roots and identifiers supposed in [5] for achieving consensus, approximate consensus can be solved even though roots arbitrarily change over time and processes are anonymous. In these respects, approximate consensus seems to be more suitable than consensus for handling real network dynamicity.

While anonymity of processes does not affect solvability, it could increase decision times in view of our quadratic time approximate consensus algorithm for a coordinated network with process identifiers, and the upper bound for averaging algorithms that we proved to be necessarily exponential.

Acknowledgments. We wish to thank Alex Olshevsky for helpful discussions on consensus sets, Martin Perner for many detailed comments, and Martin Biely for pointing out the full implications of the round translation in Lemma 4.

References

1. Angluin, D., Fischer, M.J., Jiang, H.: Stabilizing consensus in mobile networks. In: Gibbons, P.B., Abdelzaher, T., Aspnes, J., Rao, R. (eds.) 2nd IEEE International Conference on Distributed Computing in Sensor Systems, LNCS, vol. 4026, pp. 37–50. Springer, Heidelberg (2006)
2. Attiya, H., Lynch, N.A., Shavit, N.: Are wait-free algorithms fast? J. ACM 41(4), 725–763 (1994)
3. Attiya, H., Welch, J.: Distributed Computing. Wiley, Hoboken (2005)
4. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation: Numerical Methods. Athena Scientific, Belmont (1989)
5. Biely, M., Robinson, P., Schmid, U.: Agreement in directed dynamic networks. In: Even, G., Halldorsson, M.M. (eds.) 19th International Colloquium on Structural Information and Communication Complexity, LNCS, vol. 7355, pp. 73–84. Springer, Heidelberg (2012)
6. Blondel, V., Olshevshy, A.: How to decide consensus? A combinatorial necessary and sufficient condition and a proof that consensus is decidable but NP-hard. SIAM J. Control Optim. 52(5), 2707–2726 (2014)
7. Cao, M., Morse, A.S., Anderson, B.D.O.: Reaching a consensus in a dynamically changing environment: a graphical approach. SIAM J. Control Optim. 47(2), 575–600 (2008)

8. Cao, M., Morse, A.S., Anderson, B.D.O.: Reaching a consensus in a dynamically changing environment: convergence rates, measurement delays, and asynchronous events. SIAM J. Control Optim. 47(2), 601–623 (2008)
9. Charron-Bost, B.: Orientation and connectivity based criteria for asymptotic consensus (2013), arXiv:1303.2043v1 [cs.DC]
10. Charron-Bost, B., Schiper, A.: The Heard-Of model: computing in distributed systems with benign faults. Distrib. Comput. 22(1), 49–71 (2009)
11. Chung, F.R.: Spectral Graph Theory. AMS, Providence (1997)
12. Coulouma, É., Godard, E.: A characterization of dynamic networks where consensus is solvable. In: Moscibroda, T., Rescigno, A.A. (eds.) 20th International Colloquium on Structural Information and Communication Complexity, LNCS, vol. 8179, pp. 24–35. Springer, Heidelberg (2013)
13. Daubechies, I., Lagarias, J.C.: Sets of matrices all infinite products of which converge. Linear Algebra Appl. 161, 227–263 (1992)
14. Dolev, D., Lynch, N.A., Pinter, S.S., Stark, E.W., Weihl, W.E.: Reaching approximate agreement in the presence of faults. J. ACM 33(2), 499–516 (1986)
15. Fekete, A.D.: Asymptotically optimal algorithms for approximate agreement. Distrib. Comput. 4(1), 9–29 (1990)
16. Fischer, M.J., Lynch, N.A., Paterson, M.S.: Impossibility of distributed consensus with one faulty process. J. ACM 32(2), 374–382 (1985)
17. Kuhn, F., Lynch, N.A., Oshman, R.: Distributed computation in dynamic networks. In: 42nd ACM Symposium on Theory of Computing. pp. 513–522. ACM, New York City (2010)
18. Kuhn, F., Moses, Y., Oshman, R.: Coordinated consensus in dynamic networks. In: 30th Annual ACM Symposium on Principles of Distributed Computing. pp. 1–10. ACM, New York City (2011)
19. Lynch, N.A.: Distributed Algorithms. Morgan Kaufmann, San Francisco (1996)
20. Olshevsky, A., Tsitsiklis, J.N.: Degree fluctuations and the convergence time of consensus algorithms (2011), arXiv:1104.0454v1 [math.OC]
21. Santoro, N., Widmayer, P.: Time is not a healer. In: Monien, B., Cori, R. (eds.) 6th Symposium on Theoretical Aspects of Computer Science, LNCS, vol. 349, pp. 304–313. Springer, Heidelberg (1989)
22. Tsitsiklis, J.N.: Problems in Decentralized Decision Making and Computation. Ph.D. thesis, Massachusetts Institute of Technology (1984)
23. Xia, W., Cao, M.: Sarymsakov matrices and their application in coordinating multiagent systems. In: 31st Chinese Control Conference, pp. 6321–6326. IEEE, New York City (2012)